

Sortierroboter

Haytham N. H. Darawish, ETIT
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—In diesem Paper handelt es sich um die Entwicklung und Konstruktion eines Sortierroboters mit LEGO Mindstorms. Im Rahmen des Seminarprojektes wurde versucht, einen Sortierroboter zu bauen, der Objekte anhand der Farbe von einem Punkt zum anderen bringt. Der Aufbau, die Funktionsweise sowie das Programmieren werden in diesem Paper erklärt. Es wird außerdem auf die Probleme, die aufgetaucht sind, eingegangen. Abschließend werden die möglichen Anwendungsbereiche und das Endergebnis des Projektes vorgestellt.

I. EINLEITUNG

Mit der Entwicklung der Technologie sind wir in einer Zeit angekommen, in der das Nutzen von programmierten Robotern erheblich angestiegen ist, wodurch der Mensch als Arbeitskraft durch diese Roboter nach und nach ersetzt wird, was dazu geführt hat und führen wird, dass manche Arbeitsstellen vollständig von Robotern besetzt werden, wodurch die Weiterentwicklung und das Ausbauen von Robotern in deren Fähigkeiten und Eigenschaften nötig werden. Der in diesem Paper vorgestellte Roboter ist ein sehr gutes und realistisches Beispiel, denn dieses Gerät kann Objekte rund um die Uhr bewegen und verfrachten, ohne einen höheren Lohn anzufordern, was die Einstellung von Menschen unnötig macht, was viele Unternehmen bevorzugen. Er muss nur von Menschen gebaut und programmiert werden.

II. VORBETRACHTUNGEN

Nachdem die Grundlagen von MATLAB gelernt wurden, war die Aufgabe für jede Gruppe, die zwei bis drei Personen hat, eine Idee zu finden, an der gearbeitet wird. Jede Gruppe hat eine Lego-Mindstorms-Kiste, die ganz viele Bauteile für ganz unterschiedliche Roboter bietet, bekommen. Die Lego-Mindstorms-Kiste enthält drei Gleichstrommotoren, Tastsensor, Ultraschallsensor, Lichtsensor, Farbsensor, Räder, Verbindungsstangen usw. sowie den NXT-Baustein. Der NXT-Baustein bildet das zentrale Bauelement jedes Roboters und ist dessen "Gehirn". Mit dem NXT-Baustein können alle zu steuernden Bauteile (Gleichstrommotoren, Tastsensor, Ultraschallsensor, Lichtsensor, Farbsensor) des Roboters per Kabel verbunden und gesteuert werden. Dafür werden in MATLAB Programme am Computer erstellt und über USB-Kabel auf den NXT-Baustein übertragen [1]. Bevor man sich auf ein Thema festgelegt hat, waren einige Aufgaben zu lösen, die zur Ansteuerung von NXT gehören. Nachdem diese Aufgaben bearbeitet wurden, kam die Zeit ein Thema für das Projekt zu finden. Die Idee war, einen Sortierroboter zu entwickeln. Dafür wurde überlegt, was für Bauteile für den Aufbau benötigt werden. Angefangen wurde mit der Konstruktion des Roboters. Für die Konstruktion

des Roboters wurden von den steuernden Bauteilen der Lego-Mindstorms-Kiste drei Motoren und ein Farbsensor verwendet. Bevor angefangen wurde, ein Programm zu schreiben, mit dem NXT-Baustein programmiert wird, wurde ein Programmablauf erstellt (siehe Abbildung 3).

III. KONSTRUKTION DES ROBOTERS

Der Roboter weist drei Bauteile auf, die Klaue, den Arm und die Basis. In Abbildung 1 ist eine schematische Darstellung des Roboters gezeigt.

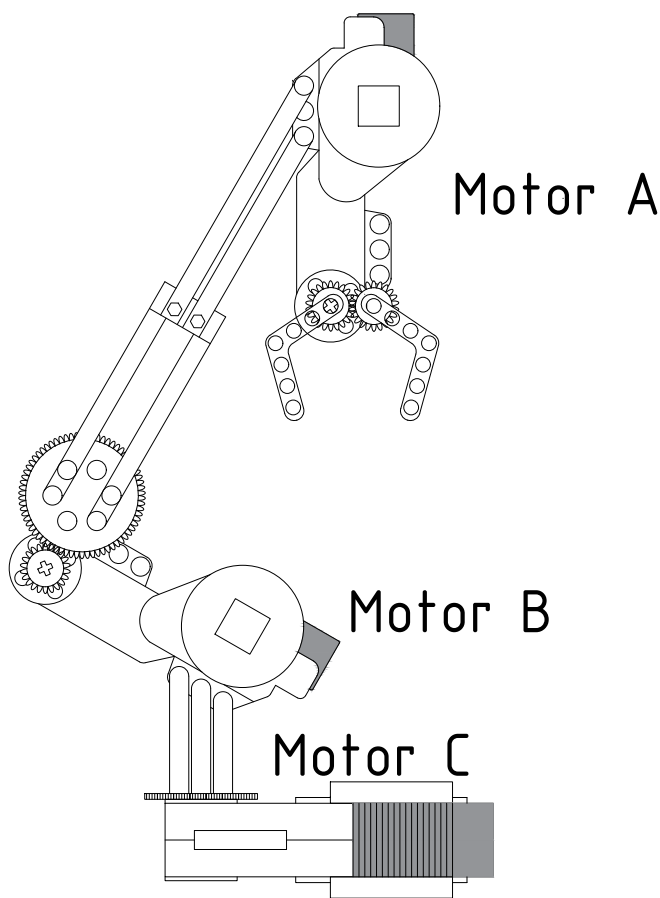


Abbildung 1. Einfache schematische Darstellung des Roboters

1) Klaue

Die Klaue ist das einfachste Bauteil des Roboters. Sie besteht grundsätzlich aus einem DC-Motor (Motor A in Abbildung 1) und einem einstufigen Stirnradgetriebe. Das einstufige Stirnradgetriebe besteht aus zwei Wellen. Das erste Zahnrad ist direkt mit der Motorwelle verbunden. Das zweite Zahnrad sitzt auf der anderen

Welle und ist zu dem ersten Zahnrad parallel gerichtet. Die zwei Zahnräder greifen also ineinander. Bei Drehung des Motors A dreht sich auch das mit der Motorwelle verbundene Zahnrad. Damit werden die Drehzahl und das Drehmoment des ersten Zahnrades auf das zweite Zahnrad übertragen. Außerdem ändert sich der Drehsinn der Zahnräder. Rotiert also bspw. das mit der Motorwelle verbundene Zahnrad gegen den Uhrzeigersinn, so wird sich das mit ihm verzahnte Zahnrad im Uhrzeigersinn drehen [2]. Damit lässt sich die Klaue durch Steuerung des Motors A öffnen und schließen.

2) Arm

Der Arm ist das zweite Bauteil des Roboters. Er liegt zwischen der Klaue und der Basis des Roboters und besteht aus einem Motor (Motor B Abbildung 1) und zwei Stirnradgetrieben. An die Motorwelle sind zwei kleine Zahnräder direkt angeschlossen. Die Zentralhand ist mit zwei größeren Zahnrädern verbunden, die zu den mit der Motorwelle verbundenen kleinen Zahnrädern parallel sind. Wie bei der Klaue übertragen sich hier bei Drehung des Motors B die Drehzahl und das Drehmoment der kleinen Zahnräder auf die größeren Zahnräder. Die Zentralhand lässt sich damit durch Steuerung des Motors B nach oben oder nach unten bewegen.

3) Basis

Der Hauptteil der Basis ist der Motor C, der für die Kreisbewegung von dem Arm und der Klaue zuständig ist. Weil der Arm und die Klaue schwer wiegen, war die Basis das schwerste Bauteil bei dem Aufbau des Roboters. Der Arm wurde direkt an die Motorwelle (Motor C) angeschlossen. Somit entsprach die Güte des Prozesses nicht 100% und folglich konnte das Objekt nicht immer direkt ins Glas geworfen werden. Der Roboter hat in einem von sechs Fällen das Ziel verfehlt. Zur Stabilitätssteigerung wurden zwei Gummiketten an der Basis fixiert.

IV. PROGRAMMABLAUF

Um die Steuerung per Mausklick ausführen zu können, wurde in MATLAB das in Abbildung 2 gezeigte Graphical User Interface erstellt, sodass man keine Befehle eingeben muss. Die erstellte Graphical User Interface verfügt über zwei Optionen, die "Manue" und "Automatik" genannt werden. Für die manuelle Steuerung wurde für jedes einzelne Objekt ein Programm geschrieben, sodass man überprüfen kann, ob alle Objekte richtig platziert sind. Das Gleiche gilt auch für die Gläser. Nachdem es überprüft wird, dass alle Objekte bei der manuellen Ausführung das Ziel getroffen haben, kommt der Einsatz des Farbsensors. Mithilfe des Farbsensors wurde eine Fallunterscheidung erstellt, sodass der Roboter automatisch die Objekte nach deren Farben sortiert.

Erst vor dem Starten des Programms wurden, wie schon erwähnt, die drei Gläser manuell durch das Userinterface an die richtige Stelle platziert. Das Objekt muss nach dem Starten

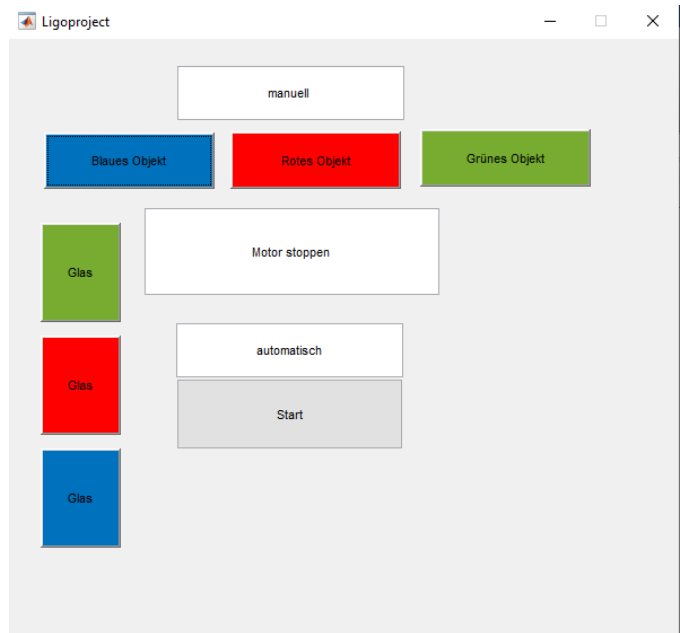


Abbildung 2. Graphical User Interface

des Programmes auf den Startpunkt, da wo der Farbsensor festgelegt ist, gestellt werden. Die Farbe wird zunächst von dem Farbsensor erkannt. Nach der Erkennung der Farbe des Objektes, wird es gegriffen und zu dem festgelegten Platz gebracht. Nun ist eine Fallunterscheidung notwendig. Ist die Farbe des Objektes Blau, dreht sich der Motor C um 90 Grad. Ist die Farbe des Objektes Rot, dreht sich der Motor C um 120 Grad. Ist die Farbe des Objektes Grün, dreht sich der Motor C um 150 Grad. Nachdem der Motor C den Arm zum festgelegten Platz gedreht hat, wird das Objekt durch Öffnen der Klaue geworfen. Dannach dreht sich der Arm wieder zum Startpunkt zurück, wo die Farbe des nächsten Objektes erkannt wird. Der Vorgang wird solange wiederholt, bis das Programm beendet wird.

V. PROBLEME

Weil nur ein an der vorderseite angelegter Farbsensor verwendet wird, muss das Objekt auf den Startpunkt gestellt werden. Das heißt, dass der Bewegungsumlauf in Bezug auf den Startpunkt ist. Dieses Problem wurde gelöst, indem die 3 Gläser durch das User Interface vor dem Starten auf den entsprechenden Stellen platziert werden. Ein weiteres Problem ist die Genauigkeit des Farbsensors. Um zu vermeiden, dass die Farbe des Objektes nicht richtig erkannt wird, muss das Objekt nah genug am Sensor sein. Das dritte Problem ist die Stabilität bei Bewegung und Greifen.

VI. VERBESSERUNG

Um das erste Problem, dass der Bewegungsumlauf in Bezug auf den Startpunkt ist, lösen zu können, muss der Aufbau des Roboters und der Programmablauf noch erweitert werden. Die umzusetzende Idee, die nicht geschafft wurde, war, dass der Roboter erst die Gläser sucht, in die die verschiedenen

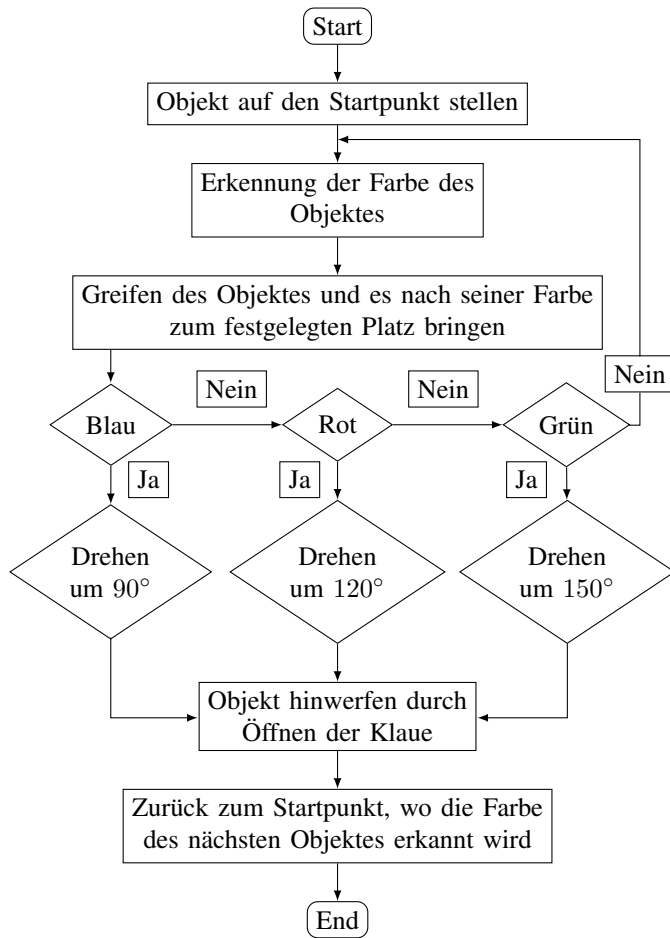


Abbildung 3. Programmablauf für die automatische Steuerung

Objekte geworfen werden. Dafür muss man weitere Sensoren verwenden.

Um die Genauigkeit des Farbsensors zu verbessern, kann man statt nur einen besser zwei Sensoren verwenden und ihre Werte vergleichen. Falls die Beiden den selben Wert ausgeben, stellt man fest, dass die Farbe des Objektes sicher erkannt wurde.

Für die Stabilitätssteigerung können mechanische Änderungen am Roboter notwendig sein.

VII. EINSATZGEBIETE

Die Sortierroboter können in zahlreichen Bereiche verwendet werden, vor allem für routinierte Aufgaben, bei denen man die gleichen Aufgaben wiederholen muss.

Im Folgenden werden einige der Einsatzgebiete genannt.

- **Landwirtschaft**
In der Landwirtschaft wird der Mechanismus des Roboters prinzipiell in Mähmaschinen eingesetzt, womit der Vorgang des Erntens durch Sensoren automatisiert wird.
- **Lagersortierung**
Um den Prozess der Lagersortierung und Kommissionie-

rung der Produkte zu optimieren, kommen solche Roboter zum Einsatz.

VIII. ENDERGEBNISS

Es wurde im Rahmen dieses Seminarprojektes geschafft, einen Sortierroboter zu entwickeln, der Objekte (kleine farbige Legosteine) durch einen Farbsensor erkennt und sie an die richtige Stelle bringt, wo mehrere Behälter (Gläser) platziert und für die drei verschiedenen Farben gedacht sind. Der Roboter hat also trotz der schon genannten Probleme seine zu erwartende Aufgabe erfüllt. Dennoch muss die Konstruktion des Roboters noch verbessert werden, da der Roboter noch nicht genug stabil ist. Mit Verwendung von mehreren und besseren Sensoren kann der Roboter die Farben ohne menschliches Eingreifen (den Legostein näher an den Sensor bringen) erkennen.

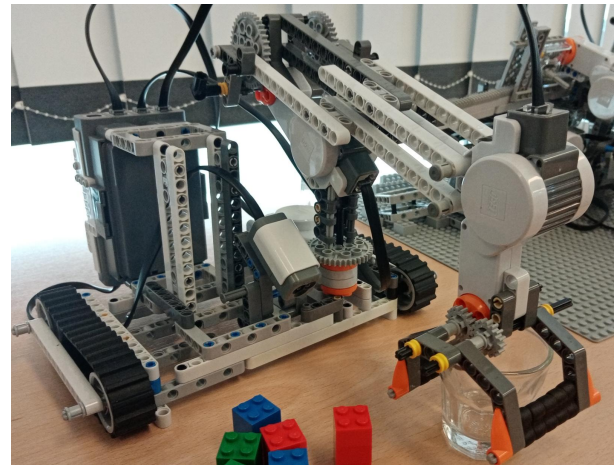


Abbildung 4. Sortierroboter von vorne

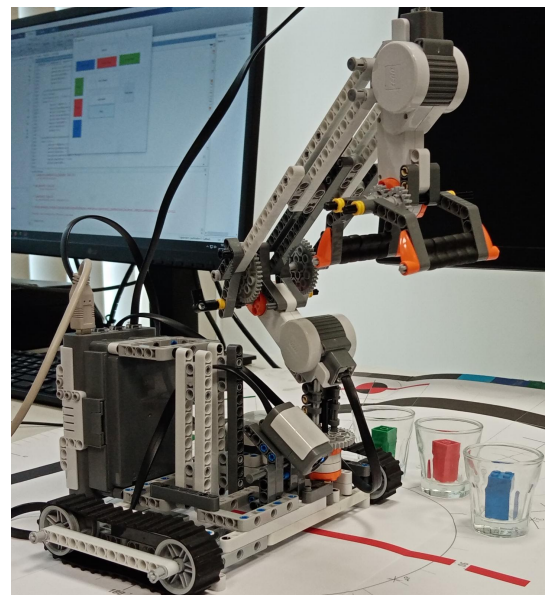


Abbildung 5. Sortierroboter von der Seite

LITERATUR

- [1] zdi Netzwerk DU.MINT Duisburg Niederrhein: LEGO Mindstorms EV3-Roboter-Sets, <https://www.uni-due.de/du-mint/lego-mindstorms>, Eingesehen am 01.03.2019
- [2] TEC-Science: Getriebetechnik, Grundlagen, Funktionsweise, <https://www.tec-science.com/de/getriebetechnik/grundlagen/funktionsweise/>, Eingesehen am 02.08.2018.

ANHANG

Während der Demonstration wurde ein Video aufgenommen und bei YouTube hochgeladen:

<https://www.youtube.com/watch?v=s7ahCUFZtL0>

Im Folgenden ist ein Beispielcode zum Schließen der Klaue gezeigt. Es wurde eine 1xn-Matrix, die im folgenden Beispielcode "Anglesss" genannt wurde, erstellt. In dieser Matrix werden die Motorwinkel registriert. Sobald die Klaue das Objekt greift, werden in der erstellten Matrix die gleichen Winkelwerte registriert, also der Winkel bleibt konstant. In diesem Fall wird die while-Schleifen mithilfe der break-Anweisung, die in Verbindung mit der if-Abfrage steht, ohne weiteres verlassen. Als nächstes wird der Motor gestoppt. Dann dreht sich der Motor nur um 1 Grad, um sicher zu stellen, dass das Objekt von der Klaue gegriffen ist.

```
COM_CloseNXT('all');
handle = COM_OpenNXT();
COM_SetDefaultNXT(handle);

disp('Klaueschliessen')
motorA = NXTMotor('A', 'Power', -50);
motorA.SpeedRegulation = false;
motorA.SmoothStart = true;
motorA.SendToNXT();

n=0;
while 1
    n=n+1;
    data = motorA.ReadFromNXT();
    Angless(n)=data.TachoCount;
    if n>3
        if Angless(end)==Angless(end-3)
            break;
        end
    end
    pause(0.05)
end
disp('Motor stoppen')
motorA.Power=0;
motorA.SendToNXT();
disp('Klaue noch mal etwas schliessen')
motorA.Power=-20;
motorA.TachoLimit = 1;
motorA.ActionAtTachoLimit = 'Holdbrake';
motorA.SendToNXT();
motorA.WaitFor()

disp('Halten')
pause(0.5)

COM_CloseNXT(handle);
```